Remarks/Arguments

A. Claims in the Case

Claims 1-39 were rejected. Claims 1-39 have been canceled. Claims 81-95 have been added. Claims 81-95 are pending in the case.

B. New Claims

Claims 81-95 have been added to the application. Support for the new claims can be found in the originally filed claims 1-39. For example, many of the features of claim 81 are described in claims 1, 9, and 10. Further support for the new claims may be found in Applicant's specification, for example, on pages 22-27

C. The Claims Are Not Obvious Over McElhiney Under 35 U.S.C. §103(a)

Claims 1-8, 12-19, and 23-30 were rejected under 35 U.S.C. §103(a) as being obvious over U.S. Patent No. 5,710,915 to McElhiney (hereinafter referred to as "McElhiney"). Applicant respectfully disagrees with these rejections.

In order to reject a claim as obvious, the Examiner has the burden of establishing a *prima facie* case of obviousness. *In re Warner et al.,* 379 F.2d 1011, 154 USPQ 173, 177-178 (CCPA 1967). To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974), MPEP § 2143.03.

New claims 81, 86, and 91 describes a combination of features including, but not limited to:

> reading a first search mask from the selected search mask table, wherein the first search mask comprises one or more search mask fields, wherein each of the one or more search mask fields corresponds to one of the one or more data element values identified in the key definition, and wherein each of the one or more search mask fields comprises a search mask field value

Applicant submits that the cited reference does not appear to teach or suggest the above-quoted feature of the claims, in combination with the other features of the claims. Applicant therefore respectfully requests removal of the 35 U.S.C. §103(a) rejections of these claims.

D. <u>The Claims Are Not Obvious Over McElhiney In View Of Kanai Under 35 U.S.C. §103(a)</u>

Claims 9-11, 20-22, and 31-39 were rejected under 35 U.S.C. §103(a) as being obvious over McElhiney in view of Kanai et al.. U.S. Patent No. 5,864,679 (hereinafter referred to as "Kanai"). Applicant respectfully disagrees with these rejections.

At least some of the features of claims 9-11, 20-22, and 31-39 have been incorporated into new claims 81, 86, and 91. New claims 81, 86, and 91 describes a combination of features including, but not limited to:

> reading a key definition from a database in response to receiving a request for a processing parameter from a first program, wherein the key definition includes the identity of one or more data element values in a set of transaction-related data;
> selecting a search mask table that corresponds to the key definition, wherein the selected search mask table comprises one or more search masks;
> reading a first search mask from the selected search mask table, wherein the first search mask comprises one or more search mask fields, wherein each of the one or more search mask fields corresponds to one of the one or more data element values identified in the key definition, and wherein each of the one or

more search mask fields comprises a search mask field value;

transferring one of the one or more data element values read from the transaction-related data to a first processing key value in response to the search mask field value indicating that the data element value from the transaction-related data is to be written to the processing key value;

transferring a wildcard value to the first processing key value in response to the search mask field value indicating that a wildcard value is to be written to the processing key value;

comparing the first processing key value to one or more key values in the database;

reading a processing parameter value from the database in response to finding a match between the first processing key value and one of the one or more key values stored in the database;

creating one or more additional processing key values if the first processing key value does not match any of the one or more key values stored in the database, wherein the one or more additional processing key values are formed using one or more additional search masks obtained from the selected search mask table;

comparing the one or more additional processing key values to one or more key values in the database until a match is found between at least one of the one or more additional processing key values;

sending the processing parameter value associated with the matching key value in the database to the first program;

wherein the processing parameter value sent to the first program is configured for use in processing the transaction-related data in the FSO computer system.

Applicant's claims are generally directed to a method of obtaining data used to process financial transactions. Generally, a financial transaction processing program may receive a transaction for processing. Upon analysis of the transaction, the processing program may send a request for processing parameters to process the transaction. The processing parameters may be accessed from a processing parameters table, which is selected based on the type of transaction received. For example, Applicant's specification states:

> Transaction processing program 500 may receive a business product transaction 502 for processing. Transaction processing program 500 may require a PCD value from PCD table C 526 for processing business product transaction 502.

> Transaction processing program may send a request 504 to key building program 506. Request 504 may include information identifying the PCD table that transaction processing program 500 requires a PCD value from.
> (Specification, pg. 22, lines 23-28)

The processing parameter table includes a plurality of processing parameters. To access the processing parameter that the received transaction needs, a key is created which is specific for the received transaction. The key will point to a particular location in the processing parameters table. The key required to access the processing parameters is created using a key building program. The key building program accesses a key definition table and obtains the key definition for the selected processing parameters table. For example, Applicant's specification states:

> Key building program 506 may include program instructions 514 for building a key value. Program instructions 514 may receive the PCD table name from request 504. In this example, the PCD table name is PCD table C. Program instructions 514 may search PCD key definition table 508 for a PCD key definition for PCD table C. PCD key definition 510 may be read from PCD key definition table 508 in response to finding the entry in the table for PCD table C.
> (Specification, pg. 23, lines 1-9)

A key is built by combining predetermined data elements from the received transaction. The key building program obtains data elements from the received transaction using one or more search masks. Each processing parameters table has an associated search mask table. The search mask table includes one or more search masks to be used by the key building program to create a key. A search mask includes one or more fields. Each of the search mask fields has a predetermined search mask field value. In some embodiments, the search mask fields may have two different values - an equal mask value and a wildcard value. For example, Applicant's specification states:

> Program instructions 514 may access search mask table 522 for PCD table C in database 520. In one embodiment of a search mask table, the search masks in the

> search mask table may be arranged in an order from first to last, wherein the search masks are read in order from first to last by a key building program until a match for a processing key value is located. After locating search mask table 522, program instructions 514 may read a first search mask 524 from search mask table 522. In one embodiment of search mask tables, each row of the search mask includes one search mask, and each search mask includes one field for each key element in the key definition associated with the search mask table. In one embodiment, wildcard mask field values and equal mask field values may be entered as mask values in search mask fields.
> (Specification, pg. 23, lines 11-20)

The search mask is used to determine what data from the received transaction is used to create a key. In one embodiment, an equal mask value may specify that the value of a data element in the received transaction is used as part of the key. A wildcard mask value may indicate that a default value may be used. By referencing the search mask and collecting the indicated data from data elements of the received transaction, the key building program may prepare a key. For example, Applicant's specification states:

> In one embodiment, an equal mask field value in a search mask field may specify that, when constructing or preparing a processing key value from the data element values in a customer account data set during processing of the customer account data set, the key element value in the processing key value corresponding to the mask field will be set to the data element value from the customer account data set. In one embodiment, a wildcard mask field value in a mask field may specify that, when constructing a processing key value from the data element values in a customer account data set during processing of the customer account data set, the key element value in the processing key value corresponding to the mask field will be set to the low collating value for the data type of the key element. For example, key elements of numeric data type may use zero (0) as a low collating value, and character fields may use spaces, or blank characters, as low collating values. Other key element types may have low collating values specific to the type.
> (Specification, pg. 23, lines 20-28)
>
> Program instructions 514 may use key definition 510 and search mask 524 to build a first key value 528 from data element values read from database 518 and

> transaction 502. Program instructions 514 may use the data elements in key definition 510 to read the data element values from the data elements. … Program instructions 514 may use search mask 524 to copy the data elements into processing key value 528.
> (Specification, pg. 24, lines 10-16)

After a key has been created, a search of the processing parameters table is made to determine if the created key matches a key entry in the processing parameters table. If a match is found, the processing parameters associated with the key are collected and transferred to the transaction processing program. For example, Applicant's specification states:

> PCD program 512 may include program instructions 516 configured for searching PCD tables and matching processing key values to PCD key values.
> (Specification, pg. 24, lines 22-23)

> Program instructions 516 may include instructions for searching the key value fields of a PCD table for a key value that matches a processing key value. In one embodiment, two key values match if they include the same key elements in the same order, and if the key element values in the first key value are the same as the key element values in the second key value for all of the key elements in the key element values.
> (Specification, pg. 25, lines 8-13)

Each processing parameters table may have associated with it a plurality of search masks. In some embodiments, these search masks may be accessed in a predetermined order. If the first created key does not match any of the key entries in the processing parameters table, one or more additional keys may be created using different search masks. To build a new key, the key building program may access the next search mask and a new key may be created based on the next search mask. The newly created key may be used to search the processing parameters table for a matching entry. This process may be repeated until a matching entry is found, or until there are no additional search masks to be used to create a new key. For example, Applicant's specification states:

> PCD program 512 may notify key building program 506 that no match was found for processing key value 528 in PCD table C 526. Referring to Figure 3b, key building program 506 may include program instructions 514 for building a key value. Program instructions 514 may use key definition 510 and a second search mask 530 read from search mask table 522 to build a second key value 532 from data element values read from database 518 and transaction 502.
> After processing key value 532 is built by program instructions 514, processing key value 532 may be passed to PCD program 512.
> (Specification, pg. 26, lines 8-14)
>
> Key building program 506 and PCD program 512 may continue to search PCD table C 526 for a match to a processing key value constructed from transaction and database data element values and wildcard values until a match is found or until all the search masks in search mask table 522 have been used without finding a matching PCD key value.
> (Specification, pg. 27, lines 13-17)

The Office Action takes the position that Kanai teaches some of the above-quoted features of independent claims 81, 86, and 91 with respect to the use of search masks. Applicant respectfully disagrees with this position.

The cited portion of Kanai states:

> the library function transmits the process ID, the data ID to be accessed, a flag indicating which library has been called up, a parameter for distinguishing a plurality of series processings defined in one process which will be described below, and parameters necessary in the access processing. The parameter for distinguishing the series processings can be given by a numerical value which is uniquely determined and assigned for each processing to be carried out in the process.
>
> The access request reception unit 201 transmits the process ID, the data ID, the parameter for distinguishing the series processings, and the flag indicating the library to the correlation information generation unit 231, while transmitting the data ID to be accessed and the parameter necessary for the data access to the data access unit 207. The data access unit 207 searches through the index in the data storage information memory unit 205 as shown in FIG. 79A by using the received

> data ID as the key. The index of FIG. 79A has a tree structure using the data ID as the key, in which the flags indicating on which processor this data is stored, at which position of which file this data is stored when this data is present on this processor, and whether the lock is set with respect to this data or not are stored in the leaves portion of the tree structure.
>
> When it is recognized that this data is present in the processor associated with this data management unit as a result of the index search, the data access unit 207 makes access to this data according to the search result, and the access result is transmitted to the library function.
> (Kanai, column 48, line 46 to column 49 line 7)

The portions of Kanai cited in the Office Action appear to teach a transaction making access to a database (Kanai, column 48, lines 32-41). A library function transmits a process ID, a data ID to be accessed, a flag indicating which library has been called up, a parameter for distinguishing a plurality of series processings, and parameters necessary in the accessing processings (Kanai, column 48, lines 46-51). A data access unit searches through an index using the data ID as a key (Kanai, column 48, lines 46-51). Kanai does not appear to teach or suggest using masks to build a key that is used to access data, as described in Applicant's claims.

The Office Action further takes the position that Kanai teaches the features of Applicant's claims related to equal and wildcard values. Applicant respectfully disagrees with this position. The portions of Kanai cited in the Office Action appear to disclose a probabilistic selection unit selecting an optimum transaction processor number (Kanai, column 31, lines 5-12). If the probabilistic selection unit does not select a processor, an arbitrary transaction processor selection unit selects an arbitrary processor. (Kanai, column 31, lines 43-46). The arbitrary processor selection unit may give a higher priority to a selection of the transaction processor predicted to have a lower load (Kanai, column 31, lines 50-62). A transaction processor is a particular computer system that is used to process transactions. The cited portion of Kanai does not appear to be related to the formation of a key for accessing data. Specifically, Kanai does not appear to teach or suggest transferring a data element value read from the transaction-related data to the processing key value in response to a search mask field value indicating that the data element value is to be written to the processing key value; and transferring a wildcard value to the processing key value in response to the search mask field value indicating that a wildcard value is to be written to the processing key value, in combination with the other features of the independent claims.

E. Additional Remarks

Based on the above, Applicant submits that the claims are now in condition for allowance. Favorable reconsideration is respectfully solicited.

Applicant respectfully requests a one-month extension of time to respond to the Office Action dated September 28, 2005. A fee authorization form including an authorization in the amount of $120.00 is enclosed for the extension of time fee. If any further extension of time is required, Applicant hereby requests the appropriate extension of time. If any fees are inadvertently omitted or if any additional fees are required or have been overpaid, please appropriately charge or credit those fees to Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C. Deposit Account Number 50-1505/5053-31301/EBM.

Respectfully submitted,

Eric B. Meyertons
Reg. No. 34,876

Attorney for Applicant

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.
P.O. BOX 398
AUSTIN, TX 78767-0398
(512) 853-8800 (voice)
(512) 853-8801 (facsimile)

Date: January 30, 2006